IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

Inventor(s):	J. P. Hearn K. N. Matthews C.C. Yu		
Case:	1-21-1		
Serial No.:	09/608,473	**Group Art Unit:**	2154
Filed:	June 30, 2000		
Examiner:	B. Barot		
Title:	MPEG Flow Identification For IP Networks		

THE COMMISSIONER OF PATENTS AND TRADEMARKS
WASHINGTON, D. C. 20231

SIR:

Appellant's Brief Under 37 C.F.R. 41.37

This is an appeal to the Board of Patent Appeals and Interferences from the Final Rejection dated July 24, 2002. Applicants are submitting this Brief in triplicate.

A Notice of Appeal was timely filed.

Real Party in Interest

The real party in interest is Lucent Technologies Inc.

Related Appeals and Interferences

There are no related appeals or interferences.

Status of Claims

Claims 7-19, 22-25, 28-30, 32, 33, and 37-39 are objected to as being dependent upon a rejected base claim but would be allowable if rewritten in independent form.

Claims 1-20, 21, 26, 27, 31, 34-36, and 40-43, which are pending in this application, stand finally rejected.

All claims are being appealed herein.

A copy of the claims as now presented are appended to this brief in the Claims Appendix attached hereto.

Status of Amendments

All amendments to the claims have been entered.

Summary of Claimed Subject Matter

All of the independent claims overcome the problem of the prior art of determining whether an internet protocol (IP) packet contains video, by actually determining that a packet contains MPEG-2 video rather than using predefined streams or priority levels that are assumed to contain such information, or any other IP packet header information, as is done in the prior art. More specifically, in accordance with an aspect of the invention, the "sync" bytes of the MPEG-2 stream are searched for within the IP packet payload, and when a pattern indicative of the sync bytes is found the sync bytes are identified and the packet is determined to contain MPEG-2 video. The sync byte was defined in MPEG-2 for over-the-air broadcasting in order that a television receiver be able to synchronize the MPEG-2 transport stream packets. Note that the MPEG-2 video, e.g., the MPEG-2 transport stream packets, may, or may not, be incorporated within real time protocol (RTP) packets before being incorporated into the IP packets. See applicants' specification, page 2, lines 20-31.

Furthermore, as explained in connection with FIG. 1, at page 5, lines 14-27, in accordance with the principles of the invention, due to the regular spacing of the sync byte, it is possible to search through UDP data payload 111 for the presence of the expected pattern of MPEG-2 video, i.e., a pattern of having a sync byte as the first byte of UDP data payload 111—after any optional RTP header—and thereafter having a sync byte at each byte position in UDP data payload 111 that is a multiple of 188. Although finding a sync byte as the first byte of UDP data payload 111, after any optional RTP header, gives a strong indication that the IP packet contains MPEG-2 video, and it is assumed that for UDP data payloads of length 188 for which the first byte has the value of a sync byte that the packet contains MPEG-2 video, preferably each potential sync byte

position should be checked. This is because the confidence level that MPEG-2 video has actually been found increases substantially when each position indeed contains a sync byte value.

FIG. 3 shows an exemplary process, in flowchart form, for processing an IP packet to determine if it contains MPEG-2 video, in accordance with the principles of the invention.

Grounds of Rejection to be Reviewed on Appeal

I. The rejection of 1-19 under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention because of the use of the phrase "may be" is the first ground of rejection to be reviewed.

II. The rejection of claims 1-6, 20-21, 26-27, 31, 34-36, and 40-43 under 35 U.S.C. 102(e) as being anticipated by United States Patent No. 6,557,031 issued to Mimura et al. on April 29, 2003 is the second ground of rejection to be reviewed.

Argument

Ground I – Rejection Under 35 U.S.C. 112, Second Paragraph

Claims 1-19 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The Office Action states that the phrase "may be" renders the claim indefinite because it is unclear whether the limitations following the phrase are part of the claimed invention.

Applicants respectfully traverse this ground of rejection for the following reason.

Applicants note that the words following "may be" are not so much limitations of the claimed invention but rather are merely descriptive of a conventional element that may, but need not, exist within an IP data payload. As such, the "may be" language is an accurate characterization of the IP data payload, in that it accurately characterizes the environment in which the invention may operate. Furthermore, as will be explained in more detail hereinbelow, whether the IP data payload contains the conventional element, is irrelevant.

More specifically, claim 1 states "the contents of said IP data payload of said IP packet exclusive of any information in any real time protocol (RTP) header which may be therein". Thus, the actual phrase of interest for the claim is "any real time protocol (RTP)

header which may be therein" in its entirety, rather than simply the word "therein" which follows the words "may be". This phrase characterizes an RTP header, which is the conventional element that may, or may not, be within the IP data payload of any particular IP packet. It is essentially the equivalent of stating "the contents of said IP data payload of said IP packet exclusive of any information in any real time protocol (RTP) header existing therein". Clearly, using such a formulation, the words "may be" have been eliminated. However, the conditionality as to whether or not an RTP header exists in the packet remains, because it actually stems from use of the word "any". Applicants believe that their recitation is somewhat clearer. Applicants note that use of the word "may", just like use of the word "or" does not necessarily render a claim indefinite.

A careful reading of the claim reveals if the IP data payload does not contain an RTP header, then of course, it cannot be employed in the step of identifying, since it is not there. Likewise, if an RTP header is included in the IP data payload, then it is not employed in the step of identifying, because the claim language excludes it from use. So, whether or not an RTP header is present in the IP data payload, the result is the same, i.e., no use is made in the identifying step of any information in any RTP header which might be in the IP data payload.

Thus, the claim language is definite, notwithstanding the present of the term "may be". Indeed, the definite meaning of the claim would be readily apparent to one of ordinary skill in the art. In support of this notion, applicants note that the European counterpart to this application has issued in Great Britain, France and Germany (1175098, 1175098, 60100204.0), as shown in the Evidence Appendix attached hereto, with a very similar claim 1 that has the same phrase "may be" as follows:

> A method for processing an internet protocol (IP) packet (101, 111), comprising the step of identifying that said packet contains motion picture expert group (MPEG)-2 video as a function of only the contents of said IP data payload (111) of said IP packet exclusive of any information in any real time protocol (RTP) header which may be within said IP data payload.

While applicants recognize that the Unite States Patent and Trademark Office does not give deference to any other patent-related body around the world, nevertheless, applicant are citing this European patent as evidence of that those ordinary skill in the art, whom the European Examiner may be considered to be, or to represent, would consider such language to be clear and definite.

Ground 2 - Rejection of Claims 1-6, 20-21, 26-27, 31, 34-36, and 40-43 Under 35 U.S.C. 102

Claims 1-6, 20-21, 26-27, 31, 34-36, and 40-43 are rejected under 35 U.S.C. 102(e) as being anticipated by United States Patent No. 6,557,031 issued to Mimura et al. on April 29, 2003.

The Office Action states that Mimura et al. teaches all the limitations of the rejected claims.

This ground of rejection is traversed for the following reasons.

Firstly, applicants note that their invention is only directed at identifying those **IP** packets that contain video. Thus, all of the sections of Mimura et al. relating to placing MPEG video from a known MPEG video source, e.g., as received from a cable television or satellite system, within an IP packet, such as column 2, lines 29-54, are totally irrelevant to applicants' invention. This is because it is know in advance that all that can come out from the known MPEG video source is MPEG video. There is, therefore, **no** need to examine such content. The fact that it is MPEG content is already known.

Furthermore, in Mimura et al. **IP** packets do **not** come out from those sources. This is because MPEG Transport stream (TS) packets are **not IP** packets. So any processing in Mimura et al. of TS packets is excluded from the scope of applicants' claims, even if it were to use the same techniques disclosed by applicants, which is not the case.

Secondly, those sections of Mimura et al. that relate to extracting MPEG video from IP packets do not teach applicants' invention. Instead, they determine that the IP packets contain MPEG video based on other techniques which do not render applicants' invention obvious.

More specifically, in Mimura et al., an IP packet is known to contain video based on address information. This can be seen, for example, from column 4, lines 52 through column 6, line 47, in which it is oft repeated that Mimura et al. assigns a correspondence from the IP address of the IP packets containing video to a PID value, which is the 13-bit packet identifiers that come after the synchronization byte in the MPEG video transport stream (TS), which is used then used to route the TS video in the MPEG video network,

e.g., a cable television or satellite system. In other words, it seems that in Mimura et al. that IP packets that contain MPEG video are identified based on information in the IP header, namely, an address, and then the PID is assigned or associated therewith. Clearly then in Mimura et al., the identification of a packet as containing video is **not** based on only information in the payload of the IP packet, as required by applicants' claims.

The sections of Mimura et al. cited in connection with claims 20-21, 31, 34-36, 40-43, namely, column 9, line 5 to column 12, line 15, do **not** teach searching through a payload of an IP packet for a pattern and indicating that the packet contains MPEG video only if the pattern is found, **nor** does it teach determining whether a payload of an IP packet has a length equal to an integral multiple of the length of an MPEG-2 transport stream packet. Rather, most of that section deals with forming an MPEG transport stream packet which include IP header information. **No** searching of IP packets is done for this purpose, and in fact IP packets don't even exist. And when, in the cited section, MPEG video in IP packets is to be extracted for conversion to transport stream packets, there is **no** searching involved. In the cited section it is assumed that the IP packets are known to contain video. Instead, as explained in subsequent sections, as well as column 4, lines 52 through column 6, line 47, this appears to be based on the IP header information of the IP packet, and **not** based on the content of the data payload of the IP packet, which seems to only contain MPEG video in PES format.

Thus, there is **no** teaching or suggestion in Mimura et al. to determine that an IP packet contains MPEG-2 data based solely on the IP data payload, exclusive of any RTP header therein.

Additionally, regarding claim 27, the cited section of Mimura et al., i.e., column 9, line 43 through column 10, line 11, does **not** teach processing the IP packet with a priority assigned for packets containing video when the indicating step indicates that the IP packet contains video. This is because that section is related to the formation of MPEG-TS signals which include IP header information, so there are **no** actual **IP** packets at this point, and so there can**not** be any processing of IP packets. Also, the cited section does not have an IP packet that was identified as having video based on only the IP data payload, because there was **no** identification of IP packets at this point, as recited in claim

27. Moreover, no language in the cited section contained therein indicates any type of priority processing. Indeed, the word "priority", or any synonym therefor, does **not** seem to appear in the cited section.

Applicants note that their claims clearly exclude any information in the RTP header from being considered in determining whether an MPEG video signal is present or not. Thus, only non-header information of any type is searched and used to determine the presence of MPEG video. However, there is no such teaching in Mimura et al.

Applicants note that claims of nearly identical wording were allowed by the EPO as European Patent No. 1175098, which has issued in Great Britain, France, and Germany. A copy of this European Patent is attached hereto in the Evidence Appendix. While applicants recognize that the United States Patent and Trademark Office does not defer to any other body, nevertheless, such an allowance is strong independent evidence that applicants' claims clearly recite a novel and not obvious invention that is adequately supported by the originally filed specification.

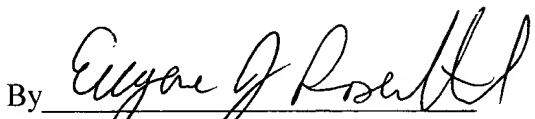
Thus, all of applicants' independent claims, and hence all of applicants' dependent claims, are allowable over Mimura et al.

Conclusion

In view of the foregoing, it is submitted that the Examiner is in error. It is, accordingly, respectfully requested that the rejection of claims 1-20, 21, 26, 27, 31, 34-36, and 40-43 be reversed, and the application passed to issue.

Respectfully,

J. P. Hearn
K. N. Matthews
C.C. Yu

By 
Eugene J. Rosenthal, Attorney
Reg. No. 36,658
732-949-1857

Lucent Technologies Inc.

Date: 6/2/05

Claims Appendix

1. (Previously presented) A method for processing an internet protocol (IP) packet, comprising the step of identifying that said packet contains motion picture expert group (MPEG)-2 video as a function of only the contents of said IP data payload of said IP packet exclusive of any information in any real time protocol (RTP) header which may be therein.

2. (original) The invention as defined in claim 1 wherein said MPEG-2 video is in transport stream format.

3. (original) The invention as defined in claim 1 wherein said IP data payload contains at least one real time protocol (RTP) packet which contains said MPEG-2 video.

4. (original) The invention as defined in claim 1 wherein said IP data payload is a unreliable datagram protocol (UDP) data payload.

5. (original) The invention as defined in claim 1 wherein said IP data payload is a transmission control protocol (TCP) data payload.

6. (original) The invention as defined in claim 1 further comprising the step of processing said IP packet with a priority assigned for packets containing video when said packet is identified in said identifying step to contain video.

7. (original) The invention as defined in claim 1 wherein said identifying step further includes the steps of:

determining whether or not there exists within said IP data payload at least an expected pattern of MPEG-2 sync bytes that is indicative of the presence of MPEG-2 video.

8. (original) The invention as defined in claim 7 wherein said determining step further comprises the step of:

comparing a first byte of said IP data payload after any real time protocol (RTP) header to the value of an MPEG-2 sync byte; and

when the result of said comparing step is that said first byte of said IP data payload has the same value as an MPEG-2 sync byte, declaring said IP packet to be an MPEG-2 packet.

9. (original) The invention as defined in claim 7 wherein said determining step further comprises the step of:

comparing a first byte of said IP data payload after any real time protocol (RTP) header to the value of an MPEG-2 sync byte; and

when the result of said comparing step is that said first byte of said IP data payload is an MPEG-2 sync byte and the length of said IP data payload after any RTP header is the same as the length of an MPEG-2 transport stream packet, declaring said IP packet to be an MPEG-2 packet.

10. (original) The invention as defined in claim 7 wherein said determining step further comprises the step of:

terminating said process and indicating that said expected pattern does not exist in said packet unless the length of said IP data payload or the length of said IP data payload less the length of a real time protocol (RTP) header is an integral multiple of the length of an MPEG-2 transport stream packet;

pointing a pointer to a byte in said IP data payload, said byte being a first byte of said IP data payload when said length of said IP data payload is an integral multiple of the length of an MPEG-2 transport stream packet and said byte being a first byte of said IP data payload after the length of a real time protocol (RTP) header when said IP data payload less the length of a real time protocol (RTP) header is an integral multiple of the length of an MPEG-2 transport stream packet;

performing a comparison between said byte being pointed to with the value of an MPEG-2 sync byte and declaring said IP packet as a candidate to be an MPEG-2 packet when the result of a most recent comparison is that said pointed to byte of said IP data payload has the same value as an MPEG-2 sync byte

adjusting said pointer to point to byte in said IP data payload that is offset toward the end of said IP packet by the length of an MPEG-2 transport stream packet;

repeating said performing and adjusting steps so long as said most recently executed performing step declared said IP packet as a candidate to be an MPEG-2 packet and the end of said IP data payload is not yet reached; and

declaring said packet to be an MPEG-2 packet when the end of said IP data payload is reached during an attempt to execute said adjusting step and said most recently executed performing step declared said IP packet as a candidate to be an MPEG-2 packet.

11. (original) The invention as defined in claim 7 wherein said expected pattern is an MPEG-2 sync byte value spaced 188 byte positions apart.

12. (original) The invention as defined in claim 7 wherein said expected pattern is an MPEG-2 sync byte value spaced apart by the length of an MPEG-2 transport stream packet.

13. (original) The invention as defined in claim 7 wherein said determining step further comprises the step of:

declaring said IP packet to be an MPEG-2 packet when a search over the length of a real time protocol header and the length of one MPEG-2 transport stream packet finds sync byte for which offset therefrom by the length of one MPEG-2 transport stream packet there is another sync byte.

14. (original) The invention as defined in claim 7 wherein said determining step further comprises the step of:

declaring said IP packet to be an MPEG-2 packet when a search over the length of a real time protocol header and the length of one MPEG-2 transport stream packet finds the value of a sync byte for which offset therefrom at each integral multiple of the length of one MPEG-2 transport stream packet there is the value of a sync byte until the end of said IP packet is reached or exeeded.

15. (original) The invention as defined in claim 7 wherein said determining step further comprises the step of:

declaring said IP packet to be an MPEG-2 packet when a search over the length of a real time protocol header and the length of one MPEG-2 transport stream packet finds the value of a sync byte for which offset therefrom by the length of one MPEG-2 transport stream packet there is the end of packet.

16. (original) The invention as defined in claim 7 wherein said each of said sync bytes has a value of 0x47.

17. (original) The invention as defined in claim 7 wherein said at least one expected pattern is the value of an MPEG-2 sync byte as the first byte of said IP data payload.

18. (original) The invention as defined in claim 7 wherein said expected pattern is the value of an MPEG-2 sync byte as the first byte of said IP data payload and every 188 bytes thereafter till the end of said IP data payload.

19. (original) The invention as defined in claim 7 wherein said at least one expected patterns includes at least one of the sets of patterns consisting of: a) the value of an MPEG-2 sync byte as the first byte of said IP data payload after the length of a real time protocol (RTP) header and said IP data payload after said RTP header length has a length of 188 bytes, b) the value of an MPEG-2 sync byte as the first byte of said IP data payload after the length of a real time protocol (RTP) header and every 188 bytes thereafter till the end of said IP data payload, c) the value of an MPEG-2 sync byte as the first byte of said IP data payload and every 188 bytes thereafter till the end of said IP data payload, d) the value of an MPEG-2 sync byte as the first byte of said IP data payload and said IP data payload has a length of 188 bytes.

20. (Previously presented) A method for processing an internet protocol (IP) packet, comprising the steps of:

searching through a payload of said IP packet exclusive of any information in any real time protocol (RTP) header therein for a pattern indicative of the presence of motion picture expert group (MPEG)-2 video; and

indicating that said IP packet contains MPEG-2 video only if said pattern is found.

21. (original) The invention as defined in claim 20 wherein said searching step further includes the step of:

determining whether a payload of said IP packet has a length equal to an integral multiple of a length of an MPEG-2 transport stream packet either before or after subtracting from said payload length the length of an RTP head.

22. (original) The invention as defined in claim 20 wherein said searching step further includes the steps of:

determining that a payload of said IP packet has a length equal to a length of an MPEG-2 transport stream packet;

comparing the value of a first byte at a first location within said packet with the value of an MPEG-2 sync byte; and

signaling said pattern is found when the result of said comparing step is that said value of said first byte at said first location matches the value of said sync byte.

23. (original) The invention as defined in claim 20 wherein said searching step further includes the step of:

determining whether a payload of said IP packet has a length equal to an integral multiple of a length of an MPEG-2 transport stream packet,

comparing the value of a first byte at a first location within said packet and each byte at an offset of a length of an MPEG-2 transport stream packet therefrom with the value of an MPEG-2 sync byte; and

signaling said pattern is found when the result of each comparison performed in said comparing step is that said first byte of said packet being compared matches the value of said sync byte.

24. (original) The invention as defined in claim 20 wherein said searching step further includes the step of:

determining whether a payload of said IP packet has a length equal to an integral multiple of a length of an MPEG-2 transport stream packet,

comparing the value of a first byte at a first location within said packet and each byte at an offset of a length of an MPEG-2 transport stream packet therefrom with the value of an MPEG-2 sync byte; and

signaling said pattern is found when the result of a majority of comparisons performed in said comparing step is that said first byte of said packet being compared matches the value of said sync byte.

25. (original) The invention as defined in claim 20 wherein said searching step further includes the step of:

determining whether a payload of said IP packet has a length equal to an integral multiple of a length of an MPEG-2 transport stream packet,

comparing the value of a first byte at a first location within said packet and each byte at an offset of a length of an MPEG-2 transport stream packet therefrom with the value of an MPEG-2 sync byte; and

signaling said pattern is found when the result of at least a majority of comparisons performed in said comparing step is that said first byte of said packet being compared matches the value of said sync byte and said packet was indicated to contain an error.

26. (original) The invention as defined in claim 20 wherein said payload is at least one of a set of payloads within an IP packet, said set consisting of: a) an IP data payload, b) an unreliable datagram protocol (UDP) data payload that does not include a real time protocol (RTP) header, c) that portion of a UDP data payload after an RTP header that is included in said UDP data payload, and d) a transmission control protocol (TCP) data payload.

27. (original) The invention as defined in claim 20 further comprising the step of processing said IP packet with a priority assigned for packets containing video when said indicating step indicates that said IP packet contains video.

28. (original) The invention as defined in claim 20 wherein said pattern corresponds to the value of an MPEG-2 sync byte regularly spaced from an initial position through the end of said payload, said regular spacing being equal to the length of an MPEG-2 transport stream packet, said initial position being located within the length of a real time protocol header and the length of one MPEG-2 transport stream packet from a start of said payload.

29. (original) The invention as defined in claim 20 wherein said pattern corresponds to the value of an MPEG-2 sync byte and the end of said IP packet spaced apart by the length of one MPEG-2 transport stream packet, said initial position being located within the length of a real time protocol header and the length of one MPEG-2 transport stream packet from a start of said payload.

30. (original) The invention as defined in claim 20 wherein said pattern corresponds to the value of an MPEG-2 sync byte regularly spaced from an initial prescribed position, said regular spacing being equal to the length of an MPEG-2 transport stream packet.

31. (Previously presented) A method for processing an internet protocol (IP) packet, comprising the steps of:

searching through a payload of said IP packet exclusive of any information in any real time protocol (RTP) header therein for a pattern indicative of the presence of motion picture expert group (MPEG)-2 video; and

indicating that said IP packet contains MPEG-2 video when said pattern is most likely found.

32. (original) The invention as defined in claim 31 wherein said pattern corresponds to an MPEG-2 sync byte regularly spaced from an initial prescribed position, said regular spacing being equal to the length of an MPEG-2 transport stream packet.

33. (original) The invention as defined in claim 32 wherein said initial prescribed position is a position within the group of positions consisting of: a) the first byte in an IP data payload of said packet, b) the first byte of a UDP payload of said IP packet, c) the first byte after an RTP header of an unreliable datagram protocol (UDP) payload of said IP packet, d) the first byte of a transport control protocol (TCP) payload of said IP packet, and e) the first byte of a TCP payload after a header contained therein indicating real time information is contained in said IP packet.

34. (original) The invention as defined in claim 31 further comprising the step of processing said IP packet with a priority assigned for packets containing video when said indicating step indicates that said IP packet contains video.

35. (Previously presented) A method for processing an internet protocol (IP) packet, comprising the steps of:

searching through a payload of said IP packet exclusive of any information in any real time protocol (RTP) header therein for a pattern indicative of video; and

indicating that said IP packet contains video when said pattern is found; and

indicating that said IP packet does not contain video when said pattern is not found.

36. (original) The invention as defined in claim 35 further comprising the step of processing said IP packet with a priority assigned for packets containing video when it is indicated that said IP packet contains video.

37. (original) The invention as defined in claim 35 wherein said pattern corresponds to the value of an MPEG-2 sync byte regularly spaced from an initial position through the end of said payload, said regular spacing being equal to the length of an MPEG-2 transport stream packet, said initial position being located within the length of a real time protocol header and the length of one MPEG-2 transport stream packet from a start of said payload.

38. (original) The invention as defined in claim 35 wherein said pattern corresponds to the value of an MPEG-2 sync byte regularly spaced from an initial prescribed position, said regular spacing being equal to the length of an MPEG-2 transport stream packet.

39. (original) The invention as defined in claim 38 wherein said initial prescribed position is a position within the group of positions consisting of: a) the first byte in an IP data payload of said packet, b) the first byte of a UDP payload of said IP packet, c) the first byte after an RTP header of an unreliable datagram protocol (UDP) payload of said IP packet, d) the first byte of a transport control protocol (TCP) payload of said IP packet, and e) the first byte of a TCP payload after a header contained therein indicating real time information is contained in said IP packet.

40. (Previously presented) A computer program in the form of machine readable instructions, said computer program being for causing a system including a processor to:

search through a payload of an internet protocol (IP) packet exclusive of any information in any real time protocol (RTP) header therein for a pattern indicative of the video;

indicate that said IP packet contains video when said pattern is found; and

indicate that said IP packet does not contain video when said pattern is not found.

41. (original) The invention as defined in claim 40 wherein said computer program further causes said processor to

process said IP packet with a priority assigned for packets containing video when said IP packet is indicated to contain video.

42. (Previously presented) Apparatus comprising:

a processor;

a memory coupled to said processor for storing a received internet protocol (IP) packet; and

a program store;

wherein said program store contains instructions for causing said processor to

search through a payload of an internet protocol (IP) packet exclusive of any information in any real time protocol (RTP) header therein for a pattern indicative of the video;

indicate that said IP packet contains video when said pattern is found; and

indicate that said IP packet does not contain video when said pattern is not found.

43. (Previously presented) Apparatus, comprising:

means for searching through a payload of an internet protocol (IP) packet exclusive of any information in any real time protocol (RTP) header therein for a pattern indicative of video; and

means for indicating that said IP packet contains video when said pattern is found.

Evidence Appendix

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 175 098 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
23.04.2003 Bulletin 2003/17

(51) Int Cl.[7]: **H04N 7/24**

(21) Application number: **01300299.3**

(22) Date of filing: **15.01.2001**

(54) MPEG flow identification for IP networks

Lokalisierung von MPEG-Flüssen für IP-Netzwerke

Identification de flux MPEG pour réseaux IP

(84) Designated Contracting States:
DE FR GB

(30) Priority: **30.06.2000 US 608473**

(43) Date of publication of application:
23.01.2002 Bulletin 2002/04

(73) Proprietor: **LUCENT TECHNOLOGIES INC.**
Murray Hill, New Jersey 07974-0636 (US)

(72) Inventors:
- **Hearn, John P.**
  New Providence, New Jersey 07974 (US)
- **Matthews, Kim N.**
  Watchung, New Jersey 07060 (US)
- **Yu, Christopher C.**
  Westfield, New Jersey 07090 (US)

(74) Representative:
Watts, Christopher Malcolm Kelway, Dr.
Lucent Technologies (UK) Ltd,
5 Mornington Road
Woodford Green Essex, IG8 0TU (GB)

(56) References cited:
EP-A- 0 914 009 **WO-A-98/02988**
WO-A-98/18233

- **A. PURI, T. CHEN: "Multimedia Systems, Standards, and Networks (chapter 18)" March 2000 (2000-03), MARCEL DEKKER XP002183848 ISBN: 082479303X * paragraph [IV.A] ***
- **D. COMER: "Internetworking with TCP/IP Vol. 1: Principles, Protocols, and Architecture (chapter 29)" 22 February 2000 (2000-02-22), PRENTICE HALL XP002183849 ISBN: 0130183806 * paragraph [29.7] * * figure 29.5 ***
- **HOFFMAN D ET AL: "RTP payload format for MPEG2/MPEG2 video" RFC 2038, 1 October 1996 (1996-10-01), XP002144280**

EP 1 175 098 B1

Description

Technical Field

[0001] This invention relates to the art of transmitting real-time-constrained information over internet protocol (IP) networks, and more particularly, to identifying particular streams of real-time-constrained information, such as video encoded using Motion Pictures Expert Group (MPEG)-2 encoding, so that they may be given appropriate processing treatment.

Background of the Invention

[0002] A problem in the art of transmitting packets containing real-time-constrained information over internet protocol (IP) networks is the need to give each type of information the appropriate processing. In order to do this, it is necessary to know the type of information that is contained within each packet as it passes through the network at each point at which the packet will be processed. In particular, video, such as using Motion Pictures Expert Group (MPEG)-2 encoded video, cannot withstand dropped packets. Therefore, in processing streams of packets that contain MPEG-2 video, it is necessary to give priority to any MPEG-2 video streams over other streams which are less sensitive, or insensitive, to dropped packets.

[0003] Prior art techniques prioritize packets based on predefined criteria without actually ascertaining, e. g., by investigating the contents of the packet, that the packets actually contain MPEG-2 video. For example, a packet flow may be defined and it is assumed that all packets in that flow are MPEG-2 packets, and packets from that flow are treated as if they contain MPEG-2 packets without regard for their actual contents. A flow is often defined by specifying source and destination IP addresses for the flow, or by specifying the source/destination port address thereof.

[0004] Another prior art technique that is used to prioritize packets is based upon the so-called "type-of-service" (TOS) byte, which is part of the IP packet header. The TOS may be used to indicate a coarse prioritization, so that, for example, it is assumed that any packet with either a predefined value in the TOS byte, or a value in the TOS byte that is at least a predefined value, contains MPEG-2 video and is treated appropriately.

[0005] Because these prior art techniques do not actually investigate the contents of the packet to be certain that it contains MPEG-2 video, it is possible that packets that do not contain MPEG-2 video will be processed as if they contained MPEG-2 video. Provided that there is sufficient bandwidth in the processing system, processing these non-MPEG-2 packets, i.e., these fake MPEG-2 packets, as if they were indeed actual MPEG-2 packets is not a problem. However, in the face of limited bandwidth-and what system does not have limited bandwidth-processing these fake MPEG-2 packets as undroppable actual MPEG-2 packets unnecessarily consumes system resources. Furthermore, these prior art prioritization arrangements can be abused by an unscrupulous user who sets his flow, or TOS byte, to indicate that he is sending MPEG-2 video, when in reality he is not.

[0006] In addition, setting up the flows in an IP network—the flows being a static configuration that specifies fixed ports—requires administration and/or reconfiguration each time a flow needs to be changed, i.e., a) when a point to point connection changes one of its points, b) when a new connection is made, or c) the like. Note that each switch or processing unit through which an MPEG-2 video stream flows must be updated with the new flow identifying information each time a flow needs to be changed. Thus, there is a constant administrative burden due to the resulting flow churn.

[0007] Another problem that arises in prior art arrangements using the TOS byte to define a flow that is to contain MPEG-2 video is the need for all of switches or processing units through which the flow passes to agree to a common TOS byte value, or set of values, that indicate MPEG-2 video. Otherwise, some switches or processing units may not properly handle, e.g., may drop, the MPEG-2 packets. It is difficult in practice to arrange for such agreement, especially when the flow travels over multiple networks.

[0008] A. PURI, T. Chen: 'Multimedia Systems, Standards, and networks' (chapter 18) March 2000 (2000-03), MARCEL DEKKER XP002183848 ISBN: 082479303X teaches delivery and control of MPEG-4 content over IP networks. The contents of a packet stream are identified by the use of header bits.

[0009] WO-A-98/18233 teaches a device and method for finding the sync pattern of a fixed-length packetized bitstream, e.g., an MPEG-2 transport stream. This is achieved using a histogram of occurrences of the sync pattern.

Summary of the Invention

[0010] A method and apparatus according to the invention are as set out in the independent claims.

[0011] We have recognized that that the problems with the prior art can be overcome, in accordance with the principles of the invention, by actually determining that a packet contains MPEG-2 video rather than using predefined streams or priority levels that are assumed to contain such information as is done in the prior art. More specifically, in accordance with an aspect of the invention, the "sync" bytes of the MPEG-2 stream are searched for within the IP packet payload, and when a pattern indicative of the sync bytes is found the sync bytes are identified and the packet is determined to contain MPEG-2 video. The sync byte was defined in MPEG-2 for over-the-air broadcasting in order that a television receiver be able to synchronize the MPEG-2 transport stream packets. Note that the MPEG-2 video,

e.g., the MPEG-2 transport stream packets, may or may not be incorporated within real time protocol (RTP) packets before being incorporated into the IP packets.

Brief Description of the Drawing

[0012] In the drawing:

FIG. I shows internet protocol (IP) packet that is of the general type for which a determination may be made as to whether or not it contains MPEG-2 video based only on the packet's IP data payload in accordance with the principles of the invention;
FIG. 2 shows an expanded version of a portion of the packet shown in FIG. 1; and
FIG. 3 shows an exemplary process, in flowchart form, for processing an IP packet to determine if it contains MPEG-2 video, in accordance with the principles of the invention.

Detailed Description

[0013] The following merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise various arrangements which, although not explicitly described or shown herein, embody the principles of the invention and are included within its spirit and scope. Furthermore, all examples and conditional language recited herein are principally intended expressly to be only for pedagogical purposes to aid the reader in understanding the principles of the invention and the concepts contributed by the inventor(s) to furthering the art, and are to be construed as being without limitation to such specifically recited examples and conditions. Moreover, all statements herein reciting principles, aspects, and embodiments of the invention, as well as specific examples thereof, are intended to encompass both structural and functional equivalents thereof. Additionally, it is intended that such equivalents include both currently known equivalents as well as equivalents developed in the future, i.e., any elements developed that perform the same function, regardless of structure.

[0014] Thus, for example, it will be appreciated by those skilled in the art that the block diagrams herein represent conceptual views of illustrative circuitry embodying the principles of the invention. Similarly, it will be appreciated that any flow charts, flow diagrams, state transition diagrams, pseudocode, and the like represent various processes which may be substantially represented in computer readable medium and so executed by a computer or processor, whether or not such computer or processor is explicitly shown.

[0015] The functions of the various elements shown in the FIGs., including functional blocks labeled as "processors", may be provided through the use of dedicated hardware as well as hardware capable of executing software in association with appropriate software. When provided by a processor, the functions may be provided by a single dedicated processor, by a single shared processor, or by a plurality of individual processors, some of which may be shared. Moreover, explicit use of the term "processor" or "controller" should not be construed to refer exclusively to hardware capable of executing software, and may implicitly include, without limitation, digital signal processor (DSP) hardware, read-only memory (ROM) for storing software, random access memory (RAM), and non-volatile storage. Other hardware, conventional and/or custom, may also be included. Similarly, any switches shown in the FIGS. are conceptual only. Their function may be carried out through the operation of program logic, through dedicated logic, through the interaction of program control and dedicated logic, or even manually, the particular technique being selectable by the implementor as more specifically understood from the context.

[0016] In the claims hereof any element expressed as a means for performing a specified function is intended to encompass any way of performing that function including, for example, a) a combination of circuit elements which performs that function or b) software in any form, including, therefore, firmware, microcode or the like, combined with appropriate circuitry for executing that software to perform the function. The invention as defined by such claims resides in the fact that the functionalities provided by the various recited means are combined and brought together in the manner which the claims call for. Applicant thus regards any means which can provide those functionalities as equivalent as those shown herein.

[0017] Unless otherwise explicitly specified herein, the drawings are not drawn to scale.

[0018] FIG. 1 shows internet protocol (IP) packet 101 that is of the general type for which a determination may be made as to whether or not it contains MPEG-2 video based only on the packet's IP data payload in accordance with the principles of the invention. More specifically, in accordance with an aspect of the invention, a search is performed for the "sync" bytes of the MPEG-2 stream within the IP packet data payload of packet 101. When a pattern indicative of the sync bytes is found, the packet is determined to contain MPEG-2 video. Otherwise, the packet is determined to contain information that is not MPEG-2 video.

[0019] Packet 101 has a series of headers which precede IP data payload 111. In particular, packet 101 contains IP header 105, which is 20 bytes long and unreliable datagram protocol/transmission control protocol (UDP/TCP) header 107, which is 8 bytes long. Note that conventionally IP header 105 and UDP/TCP header 107 are conventionally grouped together and referred to as the header for the IP packet. They are shown independently in FIG. 1 for clarity of exposition and pedagogical purposes only.

[0020] Note that IP packet 101, as shown, is a UDP packet. This is because, as of this writing, UDP is typi-

cally used for real-time streaming, as TCP requires end-to-end communication. Thus, the invention is described herein in terms of UDP packets in IP. However, those of ordinary skill in the art will readily recognize how to apply the principles of the invention to TCP packets.

[0021] IP data payload 111 follows UDP/TCP header 109. The number of bytes that are contained within IP data payload 111 is flexible, ranging from 0 and up, although certain transmission arrangements, such as ethernet, may impose other limits. Optional real time protocol (RTP) header 109, if present, is 12 bytes long, and is within IP data payload 111.

[0022] FIG. 2 shows an expanded version of IP data payload 111, which, as packet 101 is a UDP packet, is a UDP data payload. FIG. 2 shows an example in which UDP data payload 111 is carrying MPEG-2 video. As indicated, RTP header 109 may precede the MPEG-2 video within UDP data payload 111.

[0023] Consistent with any size limitations placed on it, UDP data payload 111 may carry any arbitrary number of MPEG-2 transport stream packets 201. Each of MPEG-2 transport stream packets 201 is 188 bytes in length. By virtue of the definition of the MPEG-2 transport layer, the first byte of each of MPEG-2 transport stream packets 201 is always a so-called "sync" byte 203, which has the value of 0x47.

[0024] In accordance with the principles of the invention, due to the regular spacing of the sync byte, it is possible to search through UDP data payload 111 for the presence of the expected pattern of MPEG-2 video, i.e., a pattern of having a sync byte as the first byte of UDP data payload 111—after any optional RTP header-and thereafter having a sync byte at each byte position in UDP data payload 111 that is a multiple of 188. Although finding a sync byte as the first byte of UDP data payload 111, after any optional RTP header, gives a strong indication that the IP packet contains MPEG-2 video, and it is assumed that for UDP data payloads of length 188 for which the first byte has the value of a sync byte that the packet contains MPEG-2 video, preferably each potential sync byte position should be checked. This is because the confidence level that MPEG-2 video has actually been found increases substantially when each position indeed contains a sync byte value.

[0025] In the event that most of the expected positions contain a sync byte, but one or more do not, whether or not to declare the packet as one containing MPEG-2 video is at the discretion of the implementor when designing, or configuring, the system. For example, if only one position at which a sync byte would be expected was not a sync byte, and the IP packet was indicated to have a transmission error, then the implementor may decide to have the system still treat the packet as containing MPEG-2 video.

[0026] FIG. 3 shows an exemplary process, in flowchart form, for processing an IP packet to determine if it contains MPEG-2 video, in accordance with the principles of the invention. The process is entered in step 301 when an IP packet is received. Next, in step 303, the packet is processed so there is a pointer that points to the UDP data payload within the packet, i.e., a pointer pointing within the packet is incremented to point to the UDP data payload. Thereafter, conditional branch point 305 tests to determine if the length of the UDP data payload is a multiple of 188. If the test result in step 305 is YES, indicating that there is no RTP header and that the length of the UDP data payload corresponds to a multiple of the length of MPEG-2 transport stream packet, control passes to conditional branch point 307, which tests to determine if the byte of the UDP data payload being pointed to by the pointer has the value of a sync byte, e.g., 0x47.

[0027] If the test result in step 307 is YES, control passes to step 309, which increments the pointer 188 bytes, i.e., the length of an MPEG-2 transport stream packet. This should result in the pointer pointing either at a) the beginning of the next MPEG-2 transport stream packet or to the end of the UDP data payload, which is also the end of the packet, provided the UDP data payload actually contains MPEG-2 video, or b) just a random byte when the UDP data payload does not actually contain MPEG-2 video. Thereafter, control passes to conditional branch point 311, which tests to determine if the end of the IP packet has been reached. If the test result in step 311 is NO, indicating that there yet remains additional portions of the UDP data payload to process, control passes back to step 307 to process the rest of the packet as described above. If the test result in step 311 is YES, control passes to step 313, and the IP packet is declared to be one containing MPEG-2 video, in accordance with an aspect of the invention. It may then be further processed accordingly. The process exits in step 315.

[0028] If the test result in step 307 is NO, indicating that either the first byte or another byte at a 188 multiple position does not have the value of a sync byte, control passes to step 315 and the process is exited.

[0029] If the test result in step 305 is NO, control passes to conditional branch point 317, which tests to determine if the length of the UDP data payload is equal to a multiple of 188 plus the length of the RTP header. If the test result in step 317 is YES, indicating that the UDP data payload may contain an RTP header and thereafter MPEG-2 transport stream packets, control passes to step conditional branch point 319, which tests to determine if the first bytes of the UDP data payload which correspond in length to an RTP header, have the characteristics of an RTP header, e.g., there is a video indicator in the location where the payload type field is expected. More specifically, the payload type field is 7 bits, with the definition for MPEG-2 transport stream data being 0x21.

[0030] If the test result in step 319 is NO, which indicates that there was no RTP header or that the header was not indicative of video, control passes to step 315 and the process is exited without declaring the IP packet

to be one containing MPEG-2 video. If the test result in step 319 is YES, indicating that an RTP header for video was found, control passes to step 321, in which the pointer is incremented to point to the first byte after the RTP header. Control then passes to conditional branch point 307 and the process continues as described above.

[0031] If the test result in step 317 is NO, indicating that the IP packet does not contain a whole number of MPEG-2 video transport stream packets, control passes to step 323, in which a counter variable COUNT, which is to be used as the offset into the UDP data payload, is set to 0. Next, conditional branch point 325 tests to determine if COUNT is equal to the sum of 188 and the length of an RTP header. If the test result in step 325 is YES, indicating that all the positions that could potentially contain a sync byte of a first MPEG-2 video transport stream packet in the UDP data payload have been tested and not been found to be a sync byte, the process is exited in step 315, i.e., the packet is not declared to contain MPEG-2 video. If the test result in step 325 is NO, indicating that all the positions that could potentially contain a sync byte of a first MPEG-2 video transport stream packet in the UDP data payload have not been tested, control passes to conditional branch point 327 in which another pointer, PACKTPT, is set to the value of COUNT.

[0032] Thereafter, in step 329, the byte within the UDP data payload pointed to by PACKTPT is tested to determine if it has the value of a sync byte. If the test result in step 329 is NO, indicating that the byte currently pointed to by PACKTPT is not a sync byte, control passes to step 331 in which COUNT is incremented, so that it will point to the next byte in the UDP data payload. Control then passes back to step 325, and the process continues as described above.

[0033] If the test result in step 329 is YES, indicating that the byte currently pointed to by PACKTPT indeed has the value of a sync byte, control passes to conditional branch point 333, which tests to determine if the remaining number of bytes in the packet from the place being pointed to by COUNT is less than 188. If the test result in step 333 is YES, indicating that a whole MPEG-2 video transport stream packet cannot be contained within the UDP data payload, control passes to step 315 and the process is exited, i.e., the packet is not declared to contain MPEG-2 video. If the test result in step 333 is NO, indicating that a whole MPEG-2 video transport stream packet could be contained within the UDP data payload, control passes to step 335, in which PACKTPT is incremented by 188.

[0034] At this point, if the UDP data payload indeed contains MPEG-2 video and the byte currently pointed to by COUNT is a sync byte, the byte pointed to by PACKTPT should also have the value of a sync byte, or at or beyond the end of the packet. To this end, conditional branch point 337 tests to determine if the end of the packet has been reached or is exceed. If the test result in step 337 is NO, indicating that PACKIPT points to a byte within the UDP data payload, control passes back to step 329 and the process continues as described above. If the test result in step 337 is YES, indicating that PACKTPT points to the end of the packet or beyond, control passes to step 313 and the process continues as described above.

[0035] Note that the discussion herein with regard to IP refers to IP version 4, which is essentially universally in use as of the time of the writing of this application. Those of ordinary skill in the art will be readily able to apply the principles of the invention to later developed version of IP, such as proposed version 6, should one be implemented.

Claims

1. A method for processing an internet protocol (IP) packet (101, 111), comprising the step of identifying that said packet contains motion picture expert group (MPEG)-2 video as a function of only the contents of said IP data payload (111) of said IP packet exclusive of any information in any real time protocol (RTP) header which may be within said IP data payload.

2. The method as defined in claim 1 wherein said MPEG-2 video is in transport stream format.

3. The method as defined in claim I wherein said IP data payload contains at least one RTP packet which contains said MPEG-2 video.

4. The method as defined in claim 1 wherein said IP data payload is an unreliable datagram protocol (UDP) data payload.

5. The method as defined in claim 1 wherein said IP data payload is a transmission control protocol (TCP) data payload.

6. The method as defined in claim 1 further comprising the step of processing said IP packet with a priority assigned for packets containing video when said packet is identified in said identifying step to contain video.

7. The method as defined in claim 1 wherein said identifying step further includes the steps of:

   determining whether or not there exists within said IP data payload at least an expected pattern of MPEG-2 sync bytes that is indicative of the presence of MPEG-2 video.

8. The method as defined in claim 7 wherein said determining step further comprises the step of:

comparing a first byte of said IP data payload after any real time protocol (RTP) header to the value of an MPEG-2 sync byte; and
when the result of said comparing step is that said first byte of said IP data payload has the same value as an MPEG-2 sync byte, declaring said IP packet to be an MPEG-2 packet.

9. The method as defined in claim 7 wherein said determining step further comprises the step of:

comparing a first byte of said IP data payload after any RTP header to the value of an MPEG-2 sync byte; and
when the result of said comparing step is that said first byte of said IP data payload is an MPEG-2 sync byte and the length of said IP data payload after any RTP header is the same as the length of an MPEG-2 transport stream packet, declaring said IP packet to be an MPEG-2 packet.

10. The method as defined in claim 7 wherein said determining step further comprises the step of:

terminating said process and indicating that said expected pattern does not exist in said packet unless the length of said IP data payload or the length of said IP data payload less the length of a RTP header is an integral multiple of the length of an MPEG-2 transport stream packet;
pointing a pointer to a byte in said IP data payload, said byte being a first byte of said IP data payload when said length of said IP data payload is an integral multiple of the length of an MPEG-2 transport stream packet and said byte being a first byte of said IP data payload after the length of a RTP header when said IP data payload less the length of a RTP header is an integral multiple of the length of an MPEG-2 transport stream packet;
performing a comparison between said byte being pointed to with the value of an MPEG-2 sync byte and declaring said IP packet as a candidate to be an MPEG-2 packet when the result of a most recent comparison is that said pointed to byte of said IP data payload has the same value as an MPEG-2 sync byte
adjusting said pointer to point to byte in said IP data payload that is offset toward the end of said IP packet by the length of an MPEG-2 transport stream packet;
repeating said performing and adjusting steps so long as said most recently executed performing step declared said IP packet as a candidate to be an MPEG-2 packet and the end of said IP data payload is not yet reached; and
declaring said packet to be an MPEG-2 packet when the end of said IP data payload is reached during an attempt to execute said adjusting step and said most recently executed performing step declared said IP packet as a candidate to be an MPEG-2 packet.

11. The method as defined in claim 7 wherein said expected pattern is an MPEG-2 sync byte value spaced 188 byte positions apart.

12. The method as defined in claim 7 wherein said expected pattern is an MPEG-2 sync byte value spaced apart by the length of an MPEG-2 transport stream packet.

13. The method as defined in claim 7 wherein said determining step further comprises the step of:

declaring said IP packet to be an MPEG-2 packet when a search over the length of a real time protocol header and the length of one MPEG-2 transport stream packet finds sync byte for which offset therefrom by the length of one MPEG-2 transport stream packet there is another sync byte.

14. The method as defined in claim 7 wherein said determining step further comprises the step of:

declaring said IP packet to be an MPEG-2 packet when a search over the length of a real time protocol header and the length of one MPEG-2 transport stream packet finds the value of a sync byte for which offset therefrom at each integral multiple of the length of one MPEG-2 transport stream packet there is the value of a sync byte until the end of said IP packet is reached or exceeded.

15. The method as defined in claim 7 wherein said determining step further comprises the step of:

declaring said IP packet to be an MPEG-2 packet when a search over the length of a real time protocol header and the length of one MPEG-2 transport stream packet finds the value of a sync byte for which offset therefrom by the length of one MPEG-2 transport stream packet there is the end of packet.

16. The method as defined in claim 7 wherein said each of said sync bytes has a value of 0x47.

17. The method as defined in claim 7 wherein said at least one expected pattern is the value of an MPEG-2 sync byte as the first byte of said IP data payload.

18. The method as defined in claim 7 wherein said expected pattern is the value of an MPEG-2 sync byte as the first byte of said IP data payload and every 188 bytes thereafter till the end of said IP data payload.

19. The method as defined in claim 7 wherein said at least one expected patterns includes at least one of the sets of patterns consisting of: a) the value of an MPEG-2 sync byte as the first byte of said IP data payload after the length of a real time protocol (RTP) header (109) and said IP data payload after said RTP header length has a length of 188 bytes, b) the value of an MPEG-2 sync byte as the first byte of said IP data payload after the length of a real time protocol (RTP) header and every 188 bytes thereafter till the end of said IP data payload, c) the value of an MPEG-2 sync byte as the first byte of said IP data payload and every 188 bytes thereafter till the end of said IP data payload, d) the value of an MPEG-2 sync byte as the first byte of said IP data payload and said IP data payload has a length of 188 bytes.

20. Apparatus, comprising:

means for searching through a IP data payload (111) of an internet protocol (IP) packet (101, 111) exclusive of any real time protocol (RTP) header therein for a pattern indicative of video; and
means for indicating that said IP packet contains video when said pattern is found.

Patentansprüche

1. Verfahren zum Verarbeiten eines Internet-Protokoll-Pakets (IP-Pakets) (101, 111), umfassend den Schritt des Feststellens, dass das Paket MPEG-2 Video (MPEG - "motion picture expert group") enthält, als Funktion nur des Inhalts der IP-Datennutzinformationen (111) des IP-Pakets ausschließlich der Informationen in einem beliebigen Echtzeitprotokoll-Anfangsblock ("real time protocol" - RTP), der sich in den IP-Datennutzinformationen befinden kann.

2. Verfahren nach Anspruch 1, wobei das MPEG-2 Video im Transportstromformat ist.

3. Verfahren nach Anspruch 1, wobei die IP-Datennutzinformationen mindestens ein RTP-Paket enthalten, welches das MPEG-2 Video enthält.

4. Verfahren nach Anspruch 1, wobei die IP-Datennutzinformationen unzuverlässige Datagramm-Protokoll-Datennutzinformationen ("unreliable datagram protocol" - UDP) sind.

5. Verfahren nach Anspruch 1, wobei die IP-Datennutzinformationen Übertragungsteuerungs-Protokoll-Datennutzinformationen ("transmission control protocol" - TCP) sind.

6. Verfahren nach Anspruch 1, des Weiteren umfassend den Schritt des Verarbeitens des IP-Pakets mit einer Priorität, die Paketen mit Video zugeordnet wird, wenn in dem Feststellungsschritt festgestellt wird, dass das Paket Video enthält.

7. Verfahren nach Anspruch 1, wobei der Feststellungsschritt des Weiteren die folgenden Schritte umfasst:

Bestimmen, ob in den IP-Datennutzinformationen wenigstens ein erwartetes Muster von MPEG-2 Synchronisationsbytes vorhanden ist, das auf das Vorhandensein von MPEG-2 Video hinweist.

8. Verfahren nach Anspruch 7, wobei der Bestimmungsschritt des Weiteren folgenden Schritt umfasst:

Vergleichen eines ersten Bytes der IP-Datennutzinformationen nach einem beliebigen Echtzeitprotokoll-Anfangsblock (RTP-Anfangsblock) mit dem Wert eines MPEG-2 Synchronisationsbytes; und

wenn das Ergebnis des Vergleichsschrittes ist, dass das erste Byte der IP-Datennutzinformationen denselben Wert wie ein MPEG-2 Synchronisationsbyte hat, Ausweisen des IP-Pakets als MPEG-2 Paket.

9. Verfahren nach Anspruch 7, wobei der Bestimmungsschritt des Weiteren folgenden Schritt umfasst:

Vergleichen eines ersten Bytes der IP-Datennutzinformationen nach einem beliebigen RTP-Anfangsblock mit dem Wert eines MPEG-2 Synchronisationsbytes; und wenn das Ergebnis des Vergleichsschrittes ist, dass das erste Byte der IP-Datennutzinformationen ein MPEG-2 Synchronisationsbyte ist und die Länge der IP-Datennutzinformationen nach einem beliebigen RTP-Anfangsblock gleich der Länge eines MPEG-2 Transportstrompakets ist, Ausweisen des IP-Pakets als MPEG-2 Paket.

10. Verfahren nach Anspruch 7, wobei der Bestimmungsschritt des Weiteren folgenden Schritt umfasst:

Beenden des Prozesses und Anzeigen, dass das erwartete Muster in dem Paket nicht vorhanden ist, wenn nicht die Länge der IP-Datennutzinformationen oder die Länge der IP-Datennutzinformationen minus der Länge eines RTP-Anfangsblocks ein ganzes Vielfaches der Länge eines MPEG-2 Transportstrompakets ist;

Richten eines Pointers auf ein Byte in den IP-Datennutzinformationen, wobei das Byte ein erstes Byte der IP-Datennutzinformationen ist, wenn die Länge der IP-Datennutzinformationen ein ganzes Vielfaches der Länge eines MPEG-2 Transportstrompakets ist und das Byte ein erstes Byte der IP-Datennutzinformationen nach der Länge eines RTP-Anfangsblocks ist, wenn die IP-Datennutzinformationen minus der Länge eines RTP-Anfangsblocks ein ganzes Vielfaches der Länge eines MPEG-2 Transportstrompakets ist;

Durchführen eines Vergleichs zwischen dem Byte, auf das der Pointer gerichtet ist, mit dem Wert eines MPEG-2 Synchronisationsbytes und Ausweisen des IP-Pakets als Kandidaten für ein MPEG-2 Paket, wenn das Ergebnis eines jüngsten Vergleichs ist, dass das Byte der IP-Datennutzinformationen, auf das der Pointer gerichtet ist, denselben Wert hat wie ein MPEG-2 Synchronisationsbyte;

Einstellen des Pointers, so dass er auf ein Byte in den IP-Datennutzinformationen gerichtet ist, das um die Länge eines MPEG-2 Transportstrompakets zu dem Ende des IP-Pakets versetzt ist;

Wiederholen der Durchführungs- und Einstellungsschritte, bis der zuletzt ausgeführte Durchführungsschritt das IP-Paket als Kandidaten für ein MPEG-2 Paket ausweist und das Ende der IP-Datennutzinformationen noch nicht erreicht ist; und

Ausweisen des Pakets als MPEG-2 Paket, wenn das Ende der IP-Datennutzinformationen während eines Versuchs, den Einstellungsschritt auszuführen, erreicht wird und der zuletzt ausgeführte Durchführungsschritt das IP-Paket als Kandidaten für ein MPEG-2 Paket ausweist.

11. Verfahren nach Anspruch 7, wobei das erwartete Muster ein MPEG-2 Synchronisationsbytewert in einem Abstand von 188 Bytepositionen ist.

12. Verfahren nach Anspruch 7, wobei das erwartete Muster ein MPEG-2 Synchronisationsbytewert in einem Abstand von der Länge eines MPEG-2 Transportstrompakets ist.

13. Verfahren nach Anspruch 7, wobei der Bestimmungsschritt des Weiteren folgenden Schritt umfasst:

Ausweisen des IP-Pakets als MPEG-2 Paket, wenn eine Suche über die Länge eines Echtzeitprotokoll-Anfangsblocks und die Länge eines MPEG-2 Transportstrompakets ein Synchronisationsbyte ergibt, zu dem ein anderes Synchronisationsbyte um die Länge eines MPEG-2 Transportstrompakets versetzt ist.

14. Verfahren nach Anspruch 7, wobei der Bestimmungsschritt des Weiteren folgenden Schritt umfasst:

Ausweisen des IP-Pakets als MPEG-2 Paket, wenn eine Suche über die Länge eines Echtzeitprotokoll-Anfangsblocks und die Länge eines MPEG-2 Transportstrompakets den Wert eines Synchronisationsbytes ergibt, zu dem der Wert eines Synchronisationsbytes bei jedem ganzen Vielfachen der Länge eines MPEG-2 Transportstrompakets versetzt ist, bis das Ende des IP-Pakets erreicht ist oder überschritten wird.

15. Verfahren nach Anspruch 7, wobei der Bestimmungsschritt des Weiteren folgenden Schritt umfasst:

Ausweisen des IP-Pakets als MPEG-2 Paket, wenn eine Suche über die Länge eines Echtzeitprotokoll-Anfangsblocks und die Länge eines MPEG-2 Transportstrompakets den Wert eines Synchronisationsbytes ergibt, zu dem das Ende des Pakets um die Länge eines MPEG-2 Transportstrompakets versetzt ist.

16. Verfahren nach Anspruch 7, wobei jedes der Synchronisationsbytes einen Wert von 0x47 hat.

17. Verfahren nach Anspruch 7, wobei das wenigstens eine erwartete Muster der Wert eines MPEG-2 Synchronisationsbytes als das erste Byte der IP-Datennutzinformationen ist.

18. Verfahren nach Anspruch 7, wobei das wenigstens eine erwartete Muster der Wert eines MPEG-2 Synchronisationsbytes als das erste Byte der IP-Datennutzinformationen und alle 188 Bytes danach bis zum Ende der IP-Datennutzinformationen ist.

19. Verfahren nach Anspruch 7, wobei das wenigstens

eine erwartete Muster wenigstens einen der Mustersätze enthält, bestehend aus: a) dem Wert eines MPEG-2 Synchronisationsbytes als das erste Byte der IP-Datennutzinformationen nach der Länge eines Echtzeitprotokoll-Anfangsblocks (RTP-Anfangsblocks) (109), wobei die IP-Datennutzinformationen nach der RTP-Anfangsblocklänge eine Länge von 188 Bytes haben, b) dem Wert eines MPEG-2 Synchronisationsbytes als das erste Byte der IP-Datennutzinformationen nach der Länge eines Echtzeitprotokoll-Anfangsblocks (RTP-Anfangsblocks) und alle 188 Bytes danach bis zum Ende der IP-Datennutzinformationen, c) dem Wert eines MPEG-2 Synchronisationsbytes als das erste Byte der IP-Datennutzinformationen und alle 188 Bytes danach bis zum Ende der IP-Datennutzinformationen, d) dem Wert eines MPEG-2 Synchronisationsbytes als das erste Byte der IP-Datennutzinformationen, wobei die IP-Datennutzinformationen eine Länge von 188 Bytes haben.

20. Vorrichtung, umfassend:

Mittel zum Durchsuchen von IP-Datennutzinformationen (111) eines Internet-Protokoll-Pakets (IP-Pakets) (101, 111) ausschließlich eines beliebigen Echtzeitprotokoll-Anfangsblocks (RTP-Anfangsblocks) darin nach einem Muster, das auf Video hinweist; und

Mittel zum Anzeigen, dass das IP-Paket Video enthält, wenn das Muster gefunden wird.

Revendications

1. Procédé de traitement d'un paquet de protocole Internet (IP) (101, 111), comprenant l'étape d'identification que ledit paquet contient de la vidéo selon le motion picture expert group (MPEG)-2 en fonction uniquement du contenu de ladite charge utile de données IP (111) dudit paquet IP à l'exclusion de toute information dans un quelconque en-tête de protocole en temps réel (RTP) pouvant se trouver dans ladite charge utile de données IP.

2. Procédé selon la revendication 1, dans lequel ladite vidéo MPEG-2 est en format de train de transport.

3. Procédé selon la revendication 1, dans lequel ladite charge utile de données IP contient au moins un paquet RTP qui contient ladite vidéo MPEG-2.

4. Procédé selon la revendication 1, dans lequel ladite charge utile de données IP est une charge utile de données de protocole de datagramme peu fiable (UDP).

5. Procédé selon la revendication 1, dans lequel ladite charge utile de données IP est une charge utile de données de protocole de commande de transmission (TCP).

6. Procédé selon la revendication 1, comprenant en outre l'étape de traitement dudit paquet IP avec une priorité attribuée aux paquets contenant de la vidéo quand ledit paquet est identifié dans ladite étape d'identification comme contenant de la vidéo.

7. Procédé selon la revendication 1, dans lequel ladite étape d'identification comporte en outre les étapes de :

détermination s'il existe ou non dans ladite charge utile de données IP au moins une configuration attendue d'octets de synchronisation MPEG-2 qui est indicative de la présence de vidéo MPEG-2.

8. Procédé selon la revendication 7, dans lequel ladite étape de détermination comprend en outre l'étape de :

comparaison d'un premier octet de ladite charge utile de données IP après n'importe quel entête de protocole en temps réel (RTP) à la valeur d'un octet de synchronisation MPEG-2 ; et quand le résultat de ladite étape de comparaison est que ledit premier octet de ladite charge utile de données IP a la même valeur qu'un octet de synchronisation MPEG-2, déclaration dudit paquet IP comme étant un paquet MPEG-2.

9. Procédé selon la revendication 7, dans lequel ladite étape de détermination comprend en outre l'étape de :

comparaison d'un premier octet de ladite charge utile de données IP après n'importe quel entête RTP à la valeur d'un octet de synchronisation MPEG-2 ; et
quand le résultat de ladite étape de comparaison est que ledit premier octet de ladite charge utile de données IP est un octet de synchronisation MPEG-2 et que la longueur de ladite charge utile de données IP après n'importe quel en-tête RTP est identique à la longueur d'un paquet de train de transport MPEG-2, déclaration dudit paquet IP comme étant un paquet MPEG-2.

10. Procédé selon la revendication 7, dans lequel ladite étape de détermination comprend en outre l'étape de :

terminaison dudit processus et indication que ladite configuration attendue n'existe pas dans ledit paquet à moins que la longueur de ladite charge utile de données IP ou la longueur de ladite charge utile de données IP moins la longueur d'un en-tête RTP ne soit un multiple entier de la longueur d'un paquet de train de transport MPEG-2 ;
pointage d'un pointeur vers un octet dans ladite charge utile de données IP, ledit octet étant un premier octet de ladite charge utile de données IP quand ladite longueur de ladite charge utile de données IP est un multiple entier de la longueur d'un paquet de train de transport MPEG-2 et ledit octet étant un premier octet de ladite charge utile de données IP après la longueur d'un en-tête RTP quand ladite charge utile de données IP moins la longueur d'un en-tête RTP est un multiple entier de la longueur d'un paquet de train de transport MPEG-2 ;
exécution d'une comparaison entre ledit octet faisant l'objet d'un pointage à la valeur d'un octet de synchronisation MPEG-2 et déclaration dudit paquet IP comme candidat pouvant être un paquet MPEG-2 quand le résultat d'une comparaison exécutée le plus récemment est que ledit octet faisant l'objet d'un pointage de ladite charge utile de données IP a la même valeur qu'un octet de synchronisation MPEG-2 ;
réglage dudit pointeur pour qu'il pointe sur un octet dans ladite charge utile de données IP qui est décalé vers la fin dudit paquet IP par la longueur d'un paquet de train de transport MPEG-2 ;
répétition desdites étapes d'exécution et de réglage tant que ladite étape d'exécution exécutée le plus récemment a déclaré ledit paquet IP comme candidat pouvant être un paquet MPEG-2 et que la fin de ladite charge utile de données IP n'est pas encore atteinte ; et
déclaration dudit paquet comme étant un paquet MPEG-2 quand la fin de ladite charge utile de données IP est atteinte durant une tentative d'exécution de ladite étape de réglage et que ladite étape d'exécution exécutée le plus récemment a déclaré ledit paquet IP comme candidat pouvant être un paquet MPEG-2.

11. Procédé selon la revendication 7, dans lequel ladite configuration attendue est une valeur d'octet de synchronisation MPEG-2 espacée par des positions de 188 octets.

12. Procédé selon la revendication 7, dans lequel ladite configuration attendue est une valeur d'octet de synchronisation MPEG-2 espacée par la longueur d'un paquet de train de transport MPEG-2.

13. Procédé selon la revendication 7, dans lequel ladite étape de détermination comprend en outre l'étape de :

déclaration dudit paquet IP comme étant un paquet MPEG-2 quand une recherche sur la longueur d'un en-tête de protocole en temps réel et la longueur d'un paquet de train de transport MPEG-2 trouve l'octet de synchronisation pour lequel le décalage à partir de celui-ci par la longueur d'un paquet de train de transport MPEG-2 est un autre octet de synchronisation.

14. Procédé selon la revendication 7, dans lequel ladite étape de détermination comprend en outre l'étape de :

déclaration dudit paquet IP comme étant un paquet MPEG-2 quand une recherche sur la longueur d'un en-tête de protocole en temps réel et la longueur d'un paquet de train de transport MPEG-2 trouve la valeur d'un octet de synchronisation pour lequel le décalage à partir de celui-ci à chaque multiple entier de la longueur d'un paquet de train de transport MPEG-2 est la valeur d'un octet de synchronisation jusqu'à ce que la fin dudit paquet IP soit atteinte ou dépassée.

15. Procédé selon la revendication 7, dans lequel ladite étape de détermination comprend en outre l'étape de :

déclaration dudit paquet IP comme étant un paquet MPEG-2 quand une recherche sur la longueur d'un en-tête de protocole en temps réel et la longueur d'un paquet de train de transport MPEG-2 trouve la valeur d'un octet de synchronisation pour lequel le décalage à partir de celui-ci par la longueur d'un paquet de train de transport MPEG-2 est la fin du paquet.

16. Procédé selon la revendication 7, dans lequel chaque dit octet de synchronisation a une valeur de 0x47.

17. Procédé selon la revendication 7, dans lequel ladite au moins une configuration attendue est la valeur d'un octet de synchronisation MPEG-2 comme premier octet de ladite charge utile de données IP.

18. Procédé selon la revendication 7, dans lequel ladite configuration attendue est la valeur d'un octet de synchronisation MPEG-2 comme premier octet de ladite charge utile de données IP et tous les 188 octets après cela jusqu'à la fin de ladite charge utile de données IP.

19. Procédé selon la revendication 7, dans lequel ladite au moins une configuration attendue comporte au moins un des ensembles de configurations consistant en : a) la valeur d'un octet de synchronisation MPEG-2 comme premier octet de ladite charge utile de données IP après la longueur d'un en-tête de protocole en temps réel (RTP) (109) et ladite charge utile de données IP après ladite longueur d'en-tête RTP a une longueur de 188 octets, b) la valeur d'un octet de synchronisation MPEG-2 comme premier octet de ladite charge utile de données IP après la longueur d'un en-tête de protocole en temps réel (RTP) et tous les 188 octets après cela jusqu'à la fin de ladite charge utile de données, c) la valeur d'un octet de synchronisation MPEG-2 comme premier octet de ladite charge utile de données IP et tous les 188 octets après cela jusqu'à la fin de ladite charge utile de données, d) la valeur d'un octet de synchronisation MPEG-2 comme premier octet de ladite charge utile de données IP et ladite charge utile de données IP a une longueur de 188 octets.

20. Dispositif, comprenant :

un moyen de recherche dans une charge utile de données IP (111) d'un paquet de protocole Internet (IP) (101, 111) à l'exclusion de tout en-tête de protocole en temps réel (RTP) dans celui-ci d'une configuration indicative de vidéo ; et un moyen d'indication que ledit paquet IP contient de la vidéo quand ladite configuration est trouvée.

FIG. 1

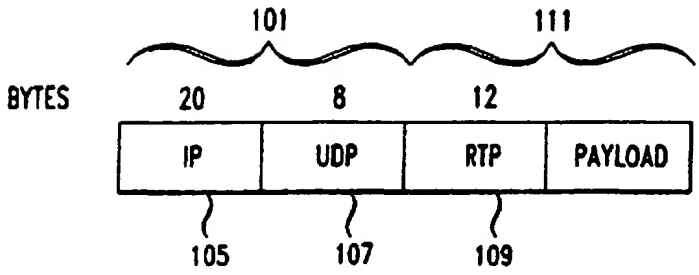

101
111
BYTES
20
8
12
IP
UDP
RTP
PAYLOAD
105
107
109

FIG. 2

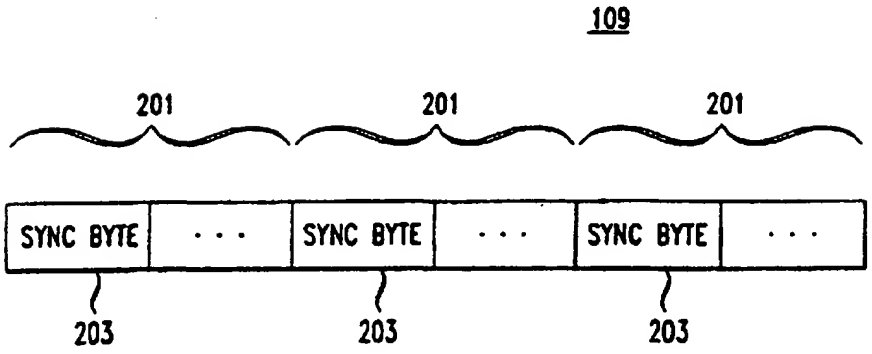
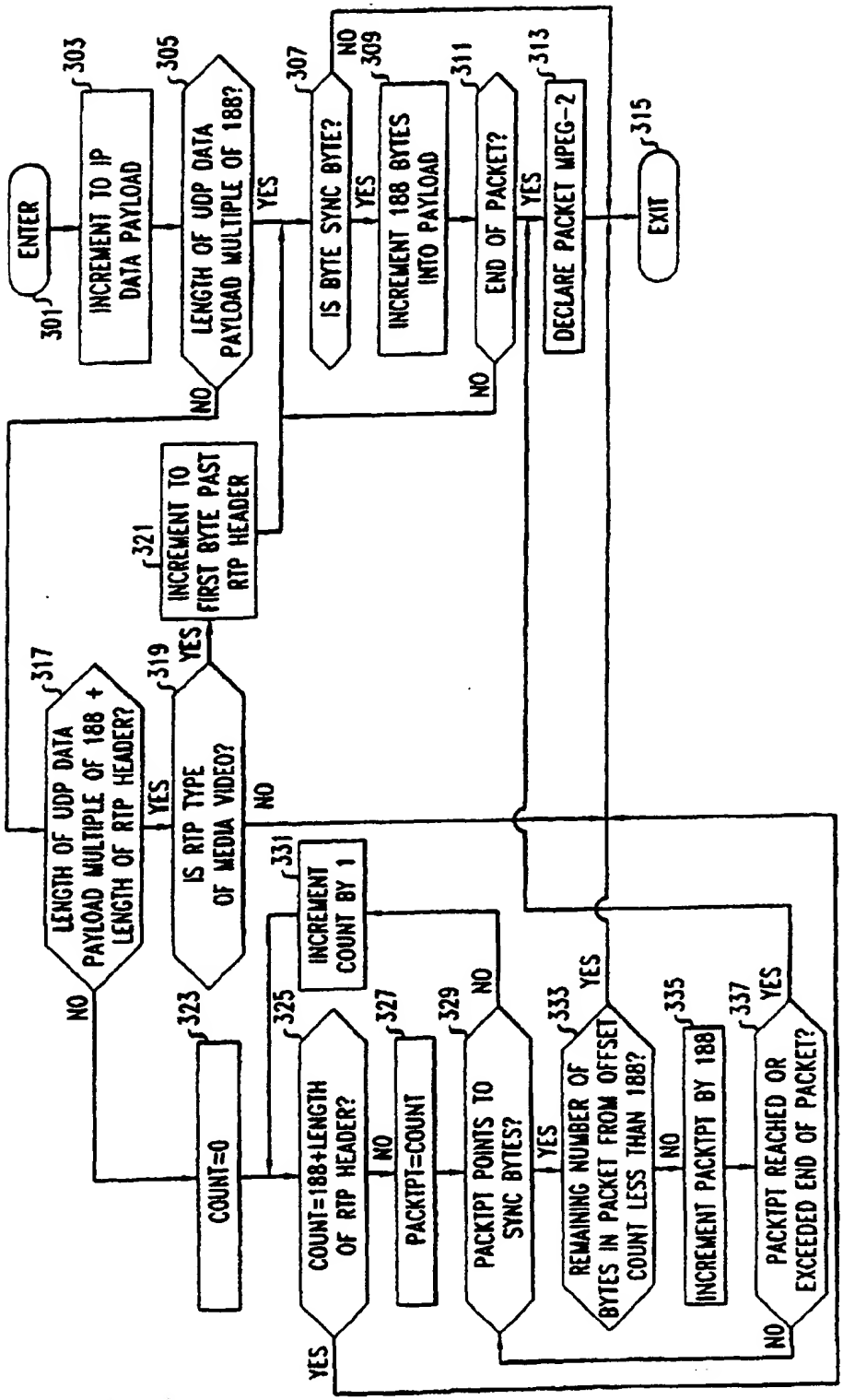

109
201
201
201
SYNC BYTE
SYNC BYTE
SYNC BYTE
203
203
203

FIG. 3


301
ENTER
303
INCREMENT TO IP DATA PAYLOAD
305
LENGTH OF UDP DATA PAYLOAD MULTIPLE OF 188?
NO
YES
307
IS BYTE SYNC BYTE?
NO
YES
309
INCREMENT 188 BYTES INTO PAYLOAD
311
END OF PACKET?
NO
YES
313
DECLARE PACKET MPEG-2
315
EXIT
317
LENGTH OF UDP DATA PAYLOAD MULTIPLE OF 188 + LENGTH OF RTP HEADER?
NO
YES
319
IS RTP TYPE OF MEDIA VIDEO?
YES
NO
321
INCREMENT TO FIRST BYTE PAST RTP HEADER
323
COUNT=0
325
COUNT=188+LENGTH OF RTP HEADER?
YES
NO
327
PACKTPT=COUNT
329
PACKTPT POINTS TO SYNC BYTES?
NO
YES
331
INCREMENT COUNT BY 1
333
REMAINING NUMBER OF BYTES IN PACKET FROM OFFSET COUNT LESS THAN 188?
YES
NO
335
INCREMENT PACKTPT BY 188
337
PACKTPT REACHED OR EXCEEDED END OF PACKET?
NO
YES